



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/823,009	04/13/2004	Michael W. Shapiro	03226/369001; SUN040527	4684
33615 7590 03/24/2008 OSHA LIANG I.L.P./SUN 1221 MCKINNEY, SUITE 2800 HOUSTON, TX 77010				
EXAMINER VU, TUAN A				
ART UNIT 2193		PAPER NUMBER		
NOTIFICATION DATE 03/24/2008		DELIVERY MODE ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

lord@oshaliang.com

hernandez@oshaliang.com

DOCKETING@OSHALIANG.COM

Office Action Summary**Application No.**

10/823,009

Applicant(s)

SHAPIRO ET AL

Examiner

Tuan A. Vu

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 4/13/04.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-23 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-23 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 4/13/04 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-946)
- 3) ☐ Information Disclosure Statement(s) (PTO/SF/08)
- Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
- Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. This action is responsive to the application filed 4/13/2004.

Claims 1-23 have been submitted for examination.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 1, 19 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 3 of U.S. Patent No. 7,305,661 (hereinafter '661).

Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following observations.

Following are but a few examples as to how the certain claims from the instant invention and from the above copending application are conflicting with each other.

As per instant claim 1, '661 claim 3 also recites 'tracing instrumented program', and 'associating with a probe' original instructions being loaded into a execution memory space. '661 does not recite 'registering a helper action with tracing', 'associating probe with a helper action', 'triggering a probe when tracing the instrumented process in a tracing framework', 'the

instrumented process' obtained from kernel-loaded instrumented application. However, as for the kernel being loaded for yielding instrumented process, '661 recites that the program being instrumented and executed is for emulating kernel; rendering the kernel loading to yield instrumented process would have been a obvious variant of '661 claim 3. As for the limitation as to "associating a helper action with a probe", as well as triggering the probe and performing the probe-associated action, '661 claim 3 recites associating a trap instruction for transfer control to a handler while executing the original instructions in the kernel. One of ordinary skill in the art would have recognized a trigger action for setting a trap action triggering context switch and association of probe with a handler as a result of the switch. It would have been obvious for one skill in the art at the time the invention was made to implement '661 probe so that it can be monitored via a trap trigger and to provide a determination based on probe association with a handler action (i.e. a action helper) enabling via this transfer control as mentioned above, perform the helper action that corresponds to said probe, based on the settings of '661. That is, '661 applying of probe with trap and control transfer to handler action is viewed as a obvious variation of the above helper action associated with the probe of the instant claim.

As per instant claim 19, '661 claim 3 also recites tracing action using probe, and associating probe with trap and handlers; that is, the rationale as to render obvious instant claim 19 regardign *helper action* in association with the *probe* and its triggering during emulation process to yield a kernel instrumented process result would have been same as set forth for instant claim 1.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 13-23 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a “useful, concrete, and tangible result” be accomplished. An “abstract idea” when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the “useful arts” when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a “useful, concrete and tangible result”.

The current focus of the Patent Office in regard to statutory inventions under 35 U.S.C. § 101 for method claims and claims that recite a judicial exception (software) is that the claimed invention recite a practical application. Practical application can be provided by a physical transformation or a useful, concrete and tangible result. The following link on the World Wide Web is the United States Patent And Trademark Office (USPTO) reference in terms of guidelines on a proper analysis on 35 U.S.C. §101 rejection.

http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelines101_20051026.pdf

Specifically, claim 13 recites a system comprising an instrumented application, helper action, and tracing framework. The framework as described in the Disclosure entails a software tool, hence as a whole, claim 13 cannot be construed as a apparatus claim comprised of hardware support of any form. The claim amounts to Functional Descriptive Material (see the USC101 Guidelines, Annex IV, pg. 52) and would not constitute any of the 4 categories of statutory subject matter. Further, listing of software entities cannot be construed as a practical application capable of yielding tangible, concrete and useful results because said recited software (application, action, framework) cannot reasonably being performed and externalized into real-

world data transformation or output via execution by computer. The claim is rejected for not leading to a statutory subject matter; and claims 14-22 are also rejected for failing to remedy to the lack of hardware support as set forth above.

Claim 23 recites a network system with nodes, comprising instrumented application, helper action, and tracing framework. These elements are perceived as mere software material and in whole claim 23 is treated as not belonging to any statutory category, and is rejected for not being able to realize the software into a practical application real-world result (i.e. Practical application can be provided by a physical transformation or a useful, concrete and tangible result).

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-2, 7, 12-14, 18-20 are rejected under 35 U.S.C. 102(b) as being anticipated by John Murayama, “Performance Profiling Using TNF”, July 2001, pg. 1-6 (hereinafter Murayama).

As per claim 1, Murayama discloses method for tracing an instrumented application, comprising:

loading the instrumented application into a kernel level (e.g. ... *inserted into the application program or kernel code* ... *TNF probes into the source code* - TNF overview, pg. 1, middle) to obtain a corresponding instrumented process (TNF execution trace - middle pg. 1);

registering a helper action (e.g. sec. **Instrumenting the Target**: ... interval around a *print* statement – pg. 2; **Interposition Libraries** – pg. 3... Example 3: “*interposed*” *tnf_unikt*, *v*, *v*) with a tracing framework;

tracing the instrumented process using the tracing framework (e.g. probes ... to trace basic kernel events such as syscalls, I/O operations, page out - TNF overview – pg. 1, middle), wherein tracing comprises

triggering a probe (e.g. **probes** ... to trace basic kernel events such as syscalls, I/O operations, page out - TNF overview – pg. 1, middle; sec **Inserting Probes**, pg. 2) in the instrumented process;

determining whether the helper action is associated with the probe (e.g. set of libraries and utilities – TNF overview pg. 1; **Interposition Libraries** – pg. 3 - Note: setting interval between start *tnf_probe* and end *tnf_probe* – Example 3, pg. 3 - to call a libraries “interposed” code reads on determining existence of association of probe and “interposed”); and

performing the helper action if the helper action is associated with the probe (e.g. Example 3: Interposed, pg. 3; Example 1: *printf* - pg. 2).

As per claims 2 and 7, Murayama discloses obtaining a helper action associated with the instrumented application (e.g. *set of libraries and utilities* – TNF overview pg. 1; **Interposition Libraries** – pg. 3 – Note: existing libraries requires obtaining a handle when performing libraries dynamic linking); and generating a helper action associated with the instrumented application (e.g. Example 3: Interposed, pg. 3; Example 1: *printf* - pg. 2)

As per claim 12, Murayama discloses wherein performing the action associated with the probe further comprises: performing a probe action (e.g. **Interposition Libraries** – pg. 3 - Note:

setting interval between start *tnf_probe* and end *tnf_probe* – Example 3, pg. 3 - to call a libraries “interposed” code reads on determining existence of association of probe and “interposed”; Example 5: *this probe enables the interposition library to trace*, pg. 3; Example 7, pg. 4) associated with the probe.

As per claim 13, Murayama discloses a system, comprising:

an instrumented application comprises a probe **probes** ... to trace basic kernel events such as syscalls, I/O operations, page out - TNF overview – pg. 1, middle; see **Inserting Probes**, pg. 2), wherein the probe is associated with an action (e.g. Example 3: Interposed, pg. 3; Example 1: *printf* - pg. 2);

a helper action associated with the instrumented application (see. **Instrumenting the Target**: ... interval around a *print* statement – pg. 2; **Interposition Libraries** – pg. 3... Example 3: “*interposed*” *tnf_unikt*, v, v); and

a tracing framework configured to trace an instrumented process (e.g. probes ... to trace basic kernel events such as syscalls, I/O operations, page out - TNF overview – pg. 1, middle) corresponding to the instrumented application and to execute the helper action (e.g. e.g. Example 3: Interposed, pg. 3; Example 1: *printf* - pg. 2; refer to claim 12) if the action is associated with the helper action.

As per claim 14, Murayama discloses wherein the helper action is generated using implementation specific details associated with the instrumented application (e.g. **Instrumenting the Target**: ... interval around a *print* statement – pg. 2; **Interposition Libraries** – pg. 3... Example 3: “*interposed*” *tnf_unikt*, v, v).

As per claims 18-19, Murayama discloses wherein the action is a generic tracing action (Example 1: *printf* - pg. 2), wherein only the helper action is executed if the helper action and the generic tracing action are associated with the probe (refer to claim 12).

As per claim 20, refer to claims 18-19 for “ wherein the helper action and the generic tracing action are executed if the helper action and the generic tracing action are associated with the probe”.

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9. Claim 23 is rejected under 35 U.S.C. 102(e) as being anticipated by Berry et al, USPN: 6,728,955 (hereinafter Berry).

As per claim 23, Berry discloses network system (e.g. Fig. 1) having a plurality of nodes, comprising:

an instrumented application comprising a probe, wherein the probe is associated with an action (e.g. hook 602- Fig. 6); a helper action associated with the instrumented application (e.g. handler - Fig. 7); and

a tracing framework configured to trace an instrumented process corresponding to the instrumented application and to execute the helper action if the action is associated with the helper action (e.g. Fig. 7);

wherein the instrumented application executes on any one of the plurality of nodes (e.g. Fig. 1, 2B, 5; col. 6, lines 49-63; col. 7, lines 10-12), wherein the helper action is located on any one of the plurality of nodes (e.g. Fig. 7), and wherein the tracing framework executes on any one of the plurality of nodes (e.g. Fig. 2B; tracing with stack unwinds – col. 6, lines 49-63; col. 7, lines 10-12).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

11. Claims 3-6, 8-11, 15, 21-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Murayama, "Performance Profiling Using TNF", July 2001.

As per claim 3, Murayama discloses linking the helper action to a specially self-describing file format source file (e.g. self-describing file format – top para, pg 2) associated (e.g. Example 1, Example 2, Example 3, pg; 2-3 – refer to claim 2 for helper libraries) with the instrumented application; and using the TNF format to initiate execution by means of a *prex* instruction (Example 6-8, pg. 3-4). But Murayama does not explicitly disclose that such TNF source file is an initialization file. In view of Murayama's mentioning of linking by the compiler because of the libraries documented from MAN page and Probe documentation file (see NPROBE; TNF_PROBE(3tnf) - pg. 3 top) the suggestion of linking a original TNF source format within an intermediate stage with additional file suggests a initial file being (as a Sun system Make facility) intermediately combined with more compiler support files or header files

known in Sun/Solaris environment (see *Solaris TNF facility* - Introduction, pg. 1). In light of this pre-execution status and nature of data defined in the TNF particular format, it would have been obvious for one skill in the art at the time the invention was made to implement the programming language and special format file by Murayama (i.e. a self-describing format file) so that this is a starting format file for initializing or defining actions as taught by Murayama, such that when linked with the probe-related libraries, this *initialization* file would serve as input to the intermediate translation format as being normally performed by a Solaris-based linking process using more files as set forth above. One would be motivated to do so because Murayama's framework thus endeavored using Solaris TNF facility should allow special language to initially set actions/definitions (e.g. as in a header file, or a input specification script) prior to linking; that is, in order to provide further readjustment, having such initial set of data prior to linking by means of standard Solaris compiling process (e.g. using a Make script or header files), such initial setting in form of file structure (e.g. as suggested via a specialized format file (e.g. as in a special format TNF script -- Murayama: Example 1, Example 2, Example 3, pg; 2-3; *tnftrace* wrapper script -- bottom pg 5) would support revising analysis after any tracing instance, and subsequent readjusting of parameter/setting (as by Murayama's tracing framework) for the probe definition or libraries association --as approached by Murayama's Solaris based compiler -- can be conveniently effectuated to help improving the process of tracing complex kernel behaviors.

As per claim 4, Murayama does not explicitly disclose wherein loading the instrumented application comprises triggering a hook in the initialization file to load the helper action into the kernel-level. But in view of Murayama use of a special format source file to initialize how

probes (Example 1, Example 2, Example 3, pg; 2-3; *infrace* wrapper script – bottom pg 5) are set to order to hook the instrumented process with the probe-related handlers, the use of initialization file to triggers the hooks would have been obvious based on the rationale set forth in claim 3.

As per claims 5-6, Murayama does not explicitly disclose wherein the helper action is stored in a process helper data structure, wherein the process helper data structure is associated with the instrumented process. But the storing of specification for probes and trigger points in a file entails a data structure. The above helper data structure would be an obvious variation of file to initialize how probe and helper actions are defined; hence would have been obvious in view of the initialization file as addressed in claim 3.

As per claim 8, Murayama does not explicitly disclose linking the helper action to an initialization file associated with the instrumented application. But based on the rationale as set forth in claim 3, and Murayama linking of the TNF file with the libraries and probe external documentation, the linking of helper action with the initialization file and instrumented application would have been obvious for the same reasons as set forth above.

As per claim 9, Murayama (in view of the rationale in claim 3) discloses wherein loading the instrumented application comprises triggering a hook (e.g. *interposed_start*, Example 3, pg 3) in the initialization file to load the helper action into the kernel-level (refer to claim 1).

As per claims 10-11, Murayama discloses wherein the helper action is stored in a process helper data structure (refer to claim 5 – Note: file inherently includes data structure to contain programmatic constructs or specification parameters); wherein the process helper data structure is associated with the instrumented process (refer to claim 6).

As per claim 15, Murayama does not explicitly disclose wherein the implementation specific details comprise at least one selected from the group consisting of an instrumented application data structure and an instrumented application algorithm. But in view of instrumented process by which Murayama initialize a special language and file structure for specifying actions and probes as set forth in claim 3, and the script file to lay algorithm for linking to additional files, libraries or header files as contemplated in Murayama's via the use of Solaris linking facility to combine TNF specification with the actions support libraries (refer to claim 3), this instrumented data structure and application algorithm falls under the ambit of a initial file with structure to coordinate an algorithmic process for linking files and libraries or macros as well-known in a Solaris compiler (e.g. using a Make facility), and would have been obvious in view of the rationale for the obviousness established in claim 3.

As per claims 21-22, Murayama does not explicitly disclose wherein the helper action is stored in a process helper data structure; wherein the process helper data structure is associated with instrumented process. But this data structure has been addressed in claim 3 and claim 15.

12. Claims 16-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Murayama, "Performance Profiling Using TNF"; and further in view of Berry et al, USPN: 6,728,955 (hereinafter Berry).

As per claims 16-17, Murayama does not explicitly disclose wherein the instrumented application data structure comprises an application stack; wherein the application stack comprises either an interpreter stack or a virtual machine stack. Sun Virtual machine using Java compiler for instrumenting code was known practice at the time the invention was made. Berry uses Sun Java compiling environment to instrument Java processes for tracing threads in part at

the kernel level (e.g. *kernel* – col. 26, lines 12-41) and teaches set up of probe (e.g. Fig. 7) to instrument virtual machine stacks (e.g. Fig. 6, Fig. 8-9, 10A-B; Fig. 12) to monitor calls at runtime. Based on needs for monitoring *virtual address* changes in memory by Murayama (see Example 8, pg. 4) when initiating probe insertions to trace memory accesses similar to Berry, it would have been obvious for one skill in the art at the time the invention was made to implement Murayama's tracing, so that if Java is main language implementation --such as taught by Berry's tracing method-- for Murayama's compilation using a Solaris environment, the runtime call stack memory within Murayama's Solaris virtual environment can be set by application data structure (as taught by Berry) comprising construct representing a JVM stack or interpreter stack as mentioned above, for the same reasons that any virtual memory access contemplated during stack usage by program must be monitored to prevent sudden conflicts (refer to Fig. 8, 10A and related text by Berry)

Conclusion

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Art Unit: 2193

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

March 14, 2008